2.0 INPUT DEVICES

2.1 POINTING DEVICE INPUT

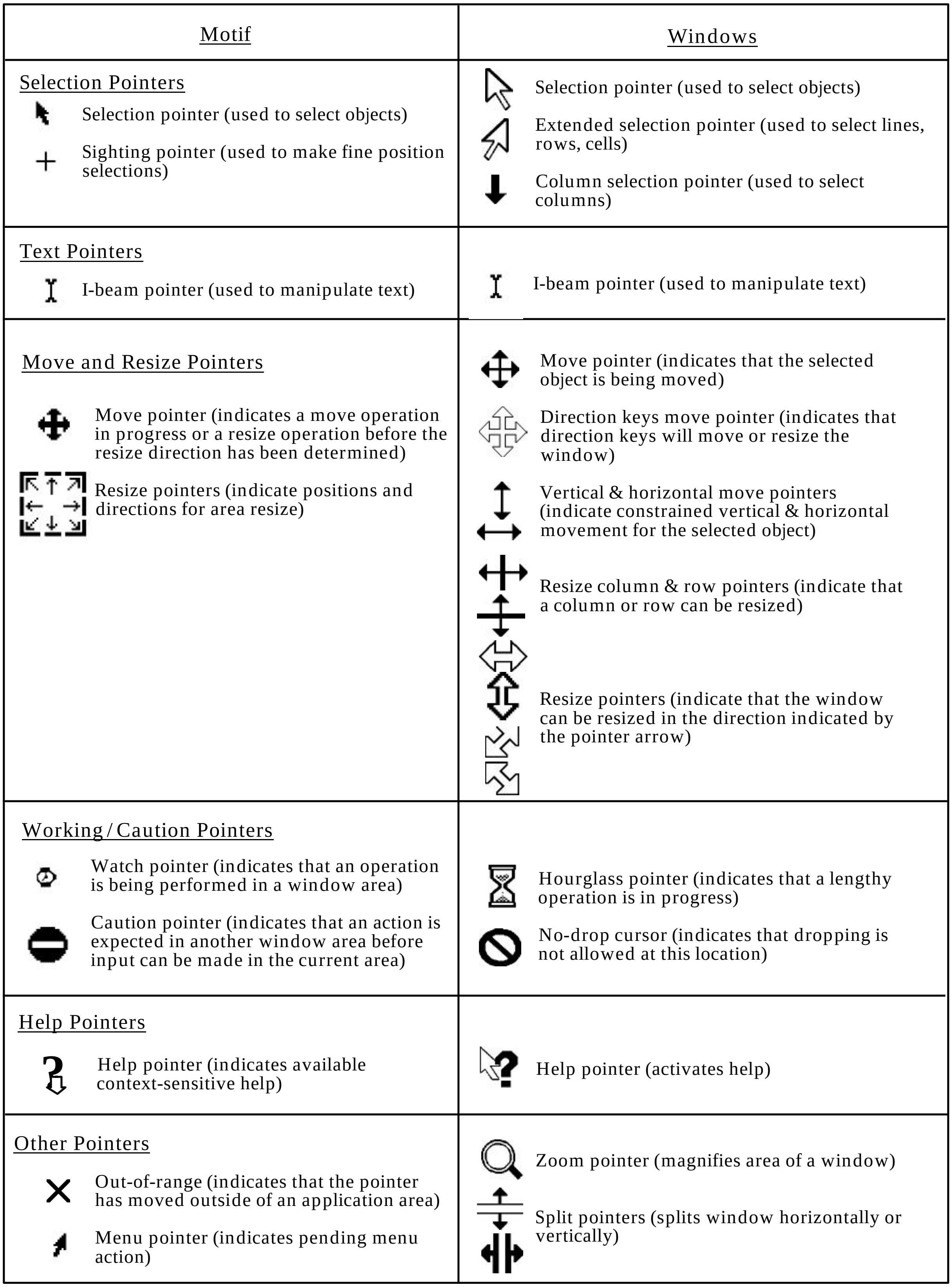
2.1.1 The Pointer

The pointing device is used to move a pointer on the screen, select objects on which the pointer is placed, and manipulate the objects directly. The pointing device is associated with a single pointer which provides a graphical representation of the location of the pointing device on the screen. The pointer may change shape depending on where it is located on the screen. The hotspot of the pointer (i.e., the active point) indicates the precise location where pointing device operations occur (e.g., the object that will be selected when users execute a select action). The location of the hotspot does not move on the screen as the pointer changes from one shape to another.

Users control the movement of the pointer by moving the pointing device; users are able to move the pointer anywhere on the screen. When users move the pointing device, the pointer moves in the corresponding direction. The position of the pointer remains where it is placed on the screen until users move it; the pointer is not moved arbitrarily by an application.

2.1.2 Pointer Shapes

The shape of the pointer depends on the functionality of the area on which the pointer is located. The pointer shapes in figure 2-1, which represent a subset of the ones defined by Motif and Windows, are to be used whenever the application provides the functionality indicated. The application can redefine the pointer shape only when the pointer is in an application window. The upper-left-pointing arrow is the general-purpose pointer for object selection in most windows. The pointer is assumed to have this shape, unless otherwise indicated.



Motif	Windows
Selection Pointers Selection pointer (used to select objects) Sighting pointer (used to make fine position selections)	Selection pointer (used to select objects) Extended selection pointer (used to select lines, rows, cells) Column selection pointer (used to select columns)
Text Pointers I-beam pointer (used to manipulate text)	I-beam pointer (used to manipulate text)
Move and Resize Pointers Move pointer (indicates a move operation in progress or a resize operation before the resize direction has been determined) Resize pointers (indicate positions and directions for area resize)	Move pointer (indicates that the selected object is being moved) Direction keys move pointer (indicates that direction keys will move or resize the window) Vertical & horizontal move pointers (indicate constrained vertical & horizontal movement for the selected object) Resize column & row pointers (indicate that a column or row can be resized) Resize pointers (indicate that the window can be resized in the direction indicated by the pointer arrow)
Working / Caution Pointers Watch pointer (indicates that an operation is being performed in a window area) Caution pointer (indicates that an action is expected in another window area before input can be made in the current area)	Hourglass pointer (indicates that a lengthy operation is in progress) No-drop cursor (indicates that dropping is not allowed at this location)
Help Pointers Help pointer (indicates available context-sensitive help)	Help pointer (activates help)
Other Pointers Out-of-range (indicates that the pointer has moved outside of an application area) Menu pointer (indicates pending menu action)	Zoom pointer (magnifies area of a window) Split pointers (splits window horizontally or vertically)

Figure 2-1. Pointer shapes in Motif and Windows.

New pointer shapes are not created for functions that already have a shape; similarly, existing shapes are not used to symbolize functions they were not designed to represent. If a new pointer shape is created for functionality not listed in figure 2-1, it is easy to see (e.g., has high contrast with the background, does not obscure other information on the screen), with a hotspot that is obvious and easy to locate. In addition, the new shape provides a hint to its purpose and is not easily confused with other objects on the screen.

2.1.3 Pointing Device Buttons

The buttons on the pointing device are used to perform various functions related to object manipulation. Because pointing devices can have different numbers of buttons, this document uses virtual button names that identify buttons by the function they perform. BSelect (i.e., the button assigned the Select function) is used to select or activate an object, BAdjust to adjust a selection, BTransfer to transfer objects, and BMenu to display pop-up menus. This style guide assumes a pointing device with three buttons is the default in Motif applications and one with two buttons in Windows applications.

> Motif Only: Two models are available for mapping pointing device buttons to virtual buttons: separate Select and Transfer functions (i.e., button 1 is used for Select and button 2 for Transfer) and integrated Select and Transfer functions (i.e., button 1 is used for both).

Table 2-1 indicates the mapping of virtual buttons to the buttons on two- and three-button pointing devices for Motif under both models and for Windows.

Table 2-1. Mapping of virtual button names to pointing device buttons in Motif and Windows.

	Motif: Integrated Select & Transfer	Motif: Separate Select & Transfer	Windows: Integrated Select & Transfer
Pointing Device with Three Buttons			
BSelect	Button 1	Button 1	Button 1
BAdjust	<Shift> + Button 1 (or Button 2 as an option)	<Shift> + Button 1	<Shift> + Button 1
BTransfer	Button 1 (or Button 2 as an option)	Button 2	Button 1 (or Button 2 as an option)
BMenu	Button 3	Button 3	Button 3
Pointing Device with Two Buttons			
BSelect	Button 1	Button 1	Button 1
BAdjust	<Shift> + Button 1	<Shift> + Button 1	<Shift> + Button 1
BTransfer	Button 1	Button 2	Button 1
BMenu	Button 2	<Alt> + Button 1	Button 2

The preferred implementation in the DII is integrated Select and Transfer functions as indicated in table 2-1.[1] In particular, the application supports the use of button 1 (i.e., the leftmost button on the pointing device) as BTransfer.

[1] Both Motif and Windows call for integrated Select and Transfer functions but allow Transfer to be assigned optionally to button 2 on the pointing device. CDE Motif gives preference to the model that integrates Select and Transfer functions while previous versions of Motif recommended the model that

Motif Only: CDE provides the option for users to assign either BAdjust or BTransfer to button 2 on a three-button pointing device when BSelect and BTransfer are integrated on button 1. When an integrated model is implemented, the application does not perform any Transfer function with button 2 when that button is used as BAdjust.

The following actions can be executed with the pointing device in both Motif and Windows:

Press	Depress and hold down a button.
Release	Release a button after it has been pressed.
Click	Press and release a button without moving the pointing device.
Double click	Press and release a button twice in rapid succession without moving the pointing device.
Move	Move the pointing device without pressing any buttons.
Drag	Move the pointing device while pressing a button.

Characteristics of the pointing device (e.g., gain and acceleration) defined at the system level (see section 7.1.6.1) are not changed by the application.

2.2 KEYBOARD INPUT

2.2.1 Fixed Function Keys

This document uses virtual keys to indicate the functions that can be executed from the keyboard. Appendix A lists the set of default functions that are defined by Motif and Windows, and appendix B maps the virtual keys for these functions to the keyboards for the standard DII hardware platforms. The application uses mappings whenever users perform any of the functions included in Appendix A.[2] If the application defines additional functions, they do not conflict with key bindings listed in this appendix.

<Ctrl>, <Shift>, and <Alt> are used only to modify the function of other keys or key combinations. <Shift>+key(s) combinations are assigned to actions that extend or are complementary to the actions of the key(s) used without <Shift>. <Ctrl>+key(s) combinations are used for infrequent actions or for actions that represent larger-scale versions of the actions assigned to the unmodified key(s). <Alt> is used only to provide access to mnemonics (see sections 3.4.2 and 5.5.5). All of the function keys defined in an application are “visible” in a window (e.g., as mnemonics and accelerators) so that users do not have to rely on memory, use on-line help, or refer to system documentation in order to interact with the application from the keyboard.

2.2.2 Variable Function Keys

It is assumed that most applications will use fixed function keys (i.e., each key has only one predefined function associated with it) to execute operations from the keyboard. However, variable function keys may also be defined for providing quick access to the operations available within the

separates Select and Transfer functions. The DII preference for the integrated model is a change in specifications from version 1.0 of the GCCS style guide.

[2] CDE Motif recommends that all application functions be available from the keyboard. While previous versions of Motif called for keyboard support for all application functionality, they, in fact, required support for all operations except range selection in text. The DoD style guide indicates that operational military systems should provide complete interchangeability but does not define a detailed model for keyboard interaction against which to assess compliance.

application. If variable function keys are used, the command names for the function keys are displayed in an application window in the form of soft keys, as shown in Figure 2-2. Soft keys can take on different meanings depending on the current state of the application; when the meaning of a key changes, the key labels displayed in the window are modified to reflect the action that will be executed if the key is used. If the application changes the functions assigned to a set of soft keys, it limits the functions to no more than two per key and includes an easy means (e.g., provide a “Home” key) for users to return to the set of base-level functions.

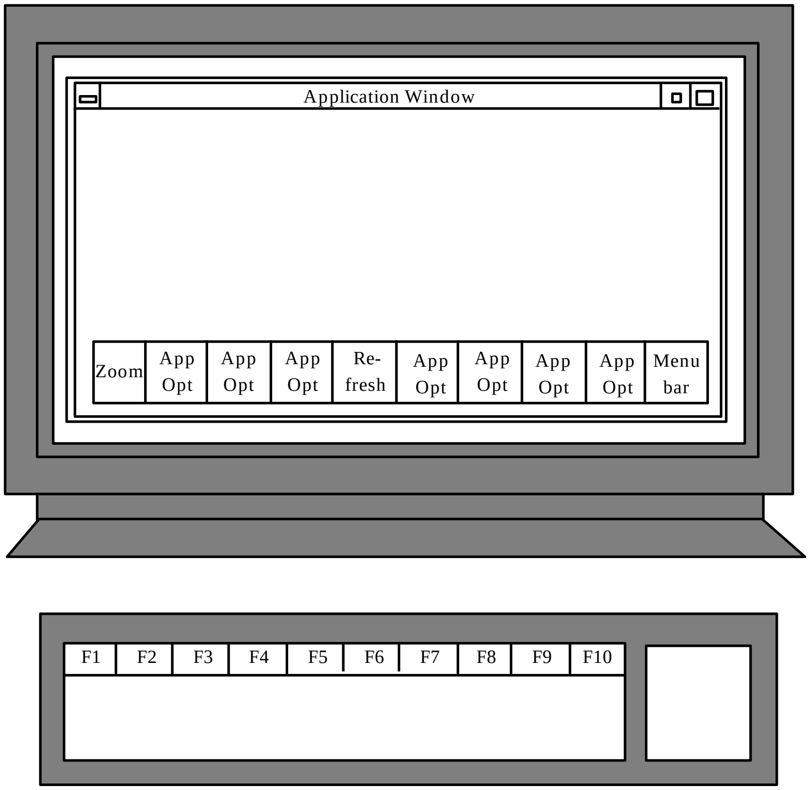


Figure 2-2. Example of soft function keys.

If variable function keys are used in the application, they are implemented in a manner consistent with input focus policy defined in section 3.1.1 (i.e., users can execute soft keys only when the window containing the keys has input focus). In addition, the actions mapped to soft keys do not conflict with the key mappings in appendix A. The keys are grayed out (to indicate their unavailability) when the window does not have focus. Developers should consider that users are likely to be working in multiple applications simultaneously, and the advantages associated with using variable function keys in the application may no longer apply when the application is one of many available to users.

2.2.3 Text Entry

2.2.3.1 Text Entry Modes

Two modes are available for text entry, with <Insert> used to toggle between the two modes. In insert mode, when users begin to type, the new character is added at the position of the text cursor. In replace mode, when users begin to type, the new character replaces the one under the text cursor.

The application provides access to both text entry modes so that users can select the mode that is more efficient given the text entry task being performed. For example, users are able to select replace mode for text entry in fields with predefined attributes (e.g., latitude/longitude and date-time group), but insert mode for free text input (e.g., the text of a message). The application does not restrict users to a single mode within a text field or arbitrarily switch between modes as users move from one field to another.

2.2.3.2 Text Entry Actions

<Space> (or <Shift><Space> in Motif) inserts a space during text entry. <Return> in Motif and <Enter> or <Ctrl><Enter> in Windows insert a carriage return in multi-line text. If no text has been selected, <Backspace> deletes the character to the left of the text cursor (backward deletion) and <Delete> deletes the character to the right of the text cursor (forward deletion). If text has been selected, both <Backspace> and <Delete> delete the selection. <Tab> inserts a tab or moves to the next tab stop in multi-line text.

Double clicking on text selects (and highlights) the word at the location of the pointer. When users highlight text and then begin typing, the text disappears, the text cursor appears, and the new text is displayed.

> Motif Only: Triple clicking on text selects (and highlights) the line of text, and quadruple clicking selects (and highlights) multiple lines (e.g., a paragraph).

2.3 ALTERNATE INPUT DEVICES

Developers considering the use of a hardware configuration that includes an alternate input device other than a mouse or trackball should submit their requests to the appropriate configuration management board for approval prior to implementation. If the use of an alternate input device is approved, the manner in which users interact with the device (e.g., for navigation and selection) is consistent with the interaction models presented in section 3 of this document.